MATHEMATICAL MODELING AND COMPUTING, Vol. 12, No. 2, pp. 603–610 (2025)

Exploration of simple graphs by a collective of agents

Stopkin A. V.

SHEI "Donbas State Pedagogical University",
13 Naukova Str., 49020, Dnipro, Ukraine

(Received 27 January 2025; Accepted 4 May 2025)

The article addresses the problem of exploring simple undirected graphs using a multi-agent system consisting of two agents: an agent-researcher, which can traverse the graph, read and modify the labels of its elements, and exchange information with the second agent—an agent-experimenter, which constructs a map of the explored graph in its memory in the form of edge and node lists. An algorithm is proposed with quadratic time, space, and communication complexities (with respect to the number of nodes in the graph). The upper estimate of the number of transitions along the edges made by the agent-researcher is estimated as $O(n^2)$. The algorithm operates using one color and one stone. An algorithm is based on depth-first traversal method.

Keywords: *graph exploration; multi-agent system; graph traversal; algorithm complexity; depth-first traversal method.*

2010 MSC: 05C85, 68R10 **DOI:** 10.23939/mmc2025.02.603

1. Introduction

A significant problem in computer science, which has received special attention, is the problem of interaction between controlling and controlled systems. One example of such interaction is the interplay between a controlling automaton, its agent, and the operational environment of that agent [1]. For instance, in [2], the interaction of these systems is represented as the movement of agents across the graph of the environment. It is clear that, in the absence of a complete model of the operational environment, targeted navigation by mobile agents is impossible. In the field of operational environment modeling, several approaches have been identified, one of which is the topological approach [3]. Under this approach, the mobile agent has access only to information about the connections between various regions of the environment, while metric and algorithmic information about the environment is unavailable. In other words, the topological model is represented as a graph with nodes, edges, and incidentors (points connecting nodes to edges) marked in various ways. Among the tasks associated with exploring environments using different agents, three primary problems are typically distinguished: the problem of agent self-localization; the problem of verifying the correspondence between the model of the environment and the actual environment, known as the map validation problem; the problem of constructing a map of the environment by mobile agents. A substantial body of research has been devoted to the task of constructing maps of the environment by mobile agents [3–10]. A number of algorithms for graph exploration have been developed, utilizing various tools of mobile agents, the formation of agent collectives, or the use of a priori information about the graph being explored, among others.

The problem of exploring unknown environments remains relevant today, as evidenced by active research in this area. However, contemporary studies focus more specifically on the operation of various multi-agent systems in exploring unknown environments and the problems that arise during their operation. Depending on the objectives, the problems of controlling multi-agent systems can be conditionally divided into the following categories: the problem of controlling a multi-agent system [11], the problem of forming a multi-agent system [12], and the problem of achieving consensus within a multi-agent system [13].

In this paper, a new graph exploration algorithm, which uses two agents, is proposed. The first agent is an agent-researcher that moves through the graph, reads and modifies marks on the graph's elements, and sends messages to the second agent. The second agent is an agent-experimenter that receives messages from the agent-researcher and, based on these messages, builds a representation of the explored graph in its memory (i.e., constructs a map of the explored graph). The algorithm is based on the depth-first traversal method.

2. Basic definitions and symbols

In this paper, connected undirected finite graphs $G = (V, E)$ without loops and multiple edges, where V is the set of nodes of the graph, and E is the set of edges (two-element subsets (u, v), where $u, v \in V$) of the graph, are considered. Triples $((u, v), v)$ are called incidentors (connection points) of edge (u, v) and node v. We denote the set of all incidentors of the graph as I. The set $L = V \cup E \cup I$ is called the set of all elements of graph G. A surjective mapping $\mu\colon L \to \{w, b\}$, where w is white and b is black, is called a coloring function of the graph G. The pair (G, μ) is called a colored graph. A sequence of pairwise adjacent nodes $u_1, u_2, \ldots, u_k$ of graph G is called a path of length k. The vicinity $Q(v)$ of node v is the set of graph elements consisting of node v, all nodes u, adjacent to v, all edges (v, u), and all incidentors $((v, u), v)$ and $((v, u), u)$. The cardinalities of the set of nodes V and the set of edges E are denoted by n and m, respectively. It is clear that $m \leqslant \frac{n(n-1)}{2}$. An isomorphism of graph G and graph H is called a bijection $\varphi\colon V_G \to V_H$, such that $(v, u) \in E_G$ if and only if $(\varphi(v), \varphi(u)) \in E_H$.

The agent-researcher is characterized by the following properties. It can move through the graph from node v to node u along the edge (v, u) that connects these nodes. In this process, the agent-researcher can color the nodes, edges, and incidentors it passes through. When the agent-researcher is at node v, it perceives the markings of all elements in the vicinity $Q(v)$ and based on this information determines the mode in which it will continue to operate, which in turn defines which edge the agent will move along next and how the graph elements will be colored. The agent-researcher has finite memory, which does not depend on the dimensionality of the explored graph.

The agent-experimenter is characterized by the following properties. It is a stationary agent located outside the graph. It receives and identifies messages from the agent-researcher and, based on these messages, constructs a representation of the explored graph in its memory in the form of edge lists and node lists. The agent-experimenter has a finite memory at each step and an unbounded internal memory that depends on the dimensionality of the explored graph.

The strategy of the agent-researcher is as follows: the agent-researcher moves depth-first as long as possible, after which it returns along its path in search of unvisited nodes and edges. If it finds such nodes, it begins moving depth-first again, and this continues until the agent returns to the starting node, at which point there are no longer any unvisited nodes and edges in its vicinity. After that, the agent-researcher finishes its task.

Thus, during its operation, the agent-researcher builds an implicit depth-first search tree, and with respect to this tree, all edges are classified into tree edges, i.e., those belonging to the constructed tree, and back edges, which do not belong to the tree.

3. Algorithm of agents' operation

Let us examine the agents' algorithms in more detail. The entire graph exploration procedure consists of two fundamentally different types of algorithms: the algorithm for exploring the graph and the computational algorithm for constructing the representation of the graph in the form of edge and node lists. It should be noted that the joint operation of these algorithms follows this sequence: the agent-researcher takes a step and sends a message to the agent-experimenter, after which the agent-experimenter processes the message, the agent-researcher takes another step, and so on.

Before the agents begin their work, all elements of the graph are white. During the graph exploration process, its elements may be colored black by the agent-researcher. For the convenience of reading

the algorithm, when describing the actions performed by the agent-researcher, we will indicate in parentheses the messages it sends to the agent-experimenter in the format "Message_A."

Algorithm of the agent-researcher's operation: At the beginning of the graph exploration algorithm, the agent-researcher is placed at an arbitrary node of the explored graph and colors it black, while the agent-experimenter immediately adds the first node to the list of nodes in its memory. The agent-researcher then moves across the white nodes, coloring these nodes, edges, and distant incidentors black, while step-by-step sending the message "Forward_A" to the agent-experimenter. Based on these messages, the agent-experimenter adds these nodes and edges to the sets V and E, respectively.

When moving forward through new nodes, the agent-researcher may encounter a white edge with white nearby and distant incidentors, and a distant black node. Such edges are called back edges, and since we do not know the number of the distant node and the quantity of such edges, there is a special procedure for exploring them. At this point, the agent stops moving forward and leaves a stone at the current node. Then, orienting itself by the black edges with the nearby black incidentor and the white distant incidentor, the agent-researcher moves backward along its path until it reaches a node where there is a white edge, the distant node of which contains the stone. At each step, the agent-researcher sends the message "Back_BE_A" to the agent-experimenter, allowing the agent-experimenter to count the number of steps the agent-researcher has moved backward. This count is necessary to determine the numbers of the unknown nodes of the back edge. As soon as the agent-researcher reaches a node where there is a stone in the vicinity, it moves across the edge that leads to this node, colors it black, and sends the message "Forward_BE_A" to the agent-experimenter. On the next step, the agent-researcher returns along this same edge, coloring both incidentors black. In this case, a message is sent to the agent-experimenter. Then, three possible scenarios can occur. (1) The current node for the agent is the starting node. This means that the agents have explored all the back edges from the node where the stone is located. In this case, the agent-researcher, orienting itself by the black edges with a white nearby and a black distant incidentor, moves forward without coloring any elements of the graph until it reaches the node where the stone is located. The agent-researcher picks up the stone and sends the message "Recognized_BE_A" to the agent-experimenter.

If the current node is not the starting node, the agent-researcher continues moving backward along its path in search of unexplored back edges. At this point, two other possibilities arise. (2) The agent-researcher reaches a node with an unexplored back edge, then the agents repeat the procedure for exploring the back edge. (3) The agent-researcher reaches a node where there are no unexplored back edges and it is not the starting node. In this case, the agent-researcher continues moving backward in search of unexplored back edges until it reaches the starting node. After that, it performs the actions described in the first case.

After completing the process of exploring back edges, the agent-researcher continues moving forward through white nodes. If, during its forward movement, the agent-researcher reaches a node in whose vicinity there are no back edges or unexplored nodes, it begins moving backward along its path in search of unexplored nodes, coloring the distant incidentors of edges black along the way. At each step, the agent sends the message "Back_A" to the agent-experimenter. Upon finding unexplored nodes, the agent-researcher resumes moving forward to explore unvisited nodes and edges.

If, during its backward movement, the agent-researcher reaches the starting node and there are no unexplored nodes in its vicinity, this means the entire graph has been explored, and the agent-researcher terminates its work, sending the message "Stop_A" to the agent-experimenter. At this moment, the agent-experimenter's memory already contains a representation of the explored graph in the form of a list of edges and nodes.

Algorithm 1 Algorithm of the agent-researcher's operation.

1: $\mu(v) := b$;
2: *if* $\exists\,(v,u) \in Q(v) \mid (\mu(v,u) = w)$ and $(\mu(u) = \mu(v) = b)$ then do;
3: $EXPL\_IE(v)$;
4: else if $\exists\,(v,u) \in Q(v) \mid (\mu(v,u) = w)$ and $(\mu(u) = w)$ then do
5: $FORWARD(v)$;
6: go to 2;
7: end do;
8: else if $\exists\,(v,u) \in Q(v) \mid (\mu((v,u),v) = b)$ and $(\mu((v,u),u) = w)$ then do
9: $BACK(v)$;
10: go to 3;
11: end do;
12: else $STOP(v)$.

$EXPL\_IE(v)$:

1. the agent leaves the stone $\mu(v) := l$;
2. if $\exists\,(v,u) \in Q(v) \mid (\mu((v,u),v) = b)$ and $(\mu(v,u) = b)$ and $(\mu((v,u),u) = w)$
3. then do
4. the agent selects $(v,u) \in Q(v) \mid (\mu((v,u),v) = b)$ and $(\mu(v,u) = b)$ and $(\mu((v,u),u) = w)$ and moves along it to the node u;
5. $v := u$;
6. the agent records in the list LM the message: $Back\_BE\_A$;
7. end do;
8. if $\exists\,(v,u) \in Q(v) \mid (\mu(v) = b)$ and $(\mu(v,u) = w)$ and $(\mu(u) = l)$
9. then do
10. the agent moves along the edge (v,u), coloring it $\mu(v,u) := b$;
11. $v := u$;
12. the agent records in the list LM the message: $Forward\_BE\_A$;
13. the agent selects $(v,u) \in Q(v) \mid (\mu((v,u),v) = w)$ and $(\mu(v,u) = b)$ and $(\mu((v,u),u) = w)$ and moves along it to the node u, coloring $\mu((v,u),v) := b$, $\mu((v,u),u) := b$;
14. go to 2 of this procedure;
15. end do;
16. else go to 2 of this procedure;
17. if $\exists\,(v,u) \in Q(v) \mid (\mu((v,u),v) = w)$ and $(\mu((v,u),u) = b)$ and $(\mu(u) = b)$
18. then do
19. the agent selects $(v,u) \in Q(v) \mid (\mu((v,u),v) = w)$ and $(\mu((v,u),u) = b)$ and $(\mu(u) = b)$ and moves along it to the node u, coloring;
20. $v := u$;
21. go to 13 of this procedure;
22. end do;
23. else do
24. the agent takes the stone $\mu(v) := b$;
25. the agent records in the list LM the message: $Recognized\_BE\_A$;
26. end do;

$FORWARD(v)$:

1. the agent selects $(v,u) \in Q(v) \mid (\mu(v,u) = w)$ and $(\mu(u) = w)$ and moves along it to the node u, coloring $\mu(v,u) := b$, $\mu((v,u),u) := b$, $\mu(u) := b$;
2. $v := u$;
3. the agent records in the list LM the message: $Forward\_A$;

$BACK(v)$:

1. the agent selects $(v,u) \in Q(v) \mid (\mu((v,u),v) = b)$ and $(\mu((v,u),u) = w)$ and moves along it to the node u, coloring $\mu((v,u),u) := b$;
2. $v := u$;
3. the agent records in the list LM the message: $Back\_A$;

$STOP(v)$:

1. the agent coloring $\mu(v) := b$;
2. the agent records in the list LM the message: $Stop\_A$;
3. the agent completes the work;

Algorithm of the agent-experimenter's operation:

Input: a list of messages LM from the agent-researcher.

*Output:*the list of nodes V_H and the list of edges E_H of the graph H, an isomorphic graph to G.

Data: V_H, E_H are the lists of nodes and edges of the graph H. ct is the counter of the number of visited nodes in graph G by the agent-researcher. i is the counter used to determine the numbers of the second nodes of marked back edges. $STOP$ is a variable used by the agent-researcher to send a message to the agent-experimenter indicating the completion of the graph exploration. $work(1), work(2), \ldots, work(t)$ is the list of node numbers along the working path, where t is the length of this list. Mes is the message being processed at the moment.

Algorithm 2 Algorithm of the agent-experimenter's operation.

1: $LM := \varnothing$; $ct := 1$, $i := 0$, $E_H := \varnothing$, $t := 1$, $work(t) := ct$, $V_H := \{1\}$, $STOP := 0$;
2: while $LM \neq \varnothing$ do
3: if $LM \neq \varnothing$ then do
4: read a message into the variable Mes;
5: $LM := LM \backslash \{Mes\}$;
6: $List\_Processing\_A()$;
7: end do;
8: end do;
9: print V_H, E_H.

$List\_processing\_A()$:

1. if $Mes =$"Forward_A" then $Forward\_A()$;
2. if $Mes =$"Back_A" then $Back\_A()$;
3. if $Mes =$"Forward_BE_A" then $Forward\_BE\_A()$;
4. if $Mes =$"Back_BE_A" then $Back\_BE\_A()$;
5. if $Mes =$"Recognized_BE_A" then $Recognized\_BE\_A()$;
6. if $Mes =$"Stop_A" then $Stop\_A()$.

$Forward\_A()$:

1. $ct := ct + 1$;
2. $t := t + 1$;
3. $work(t) := ct$;
4. $V_H := V_H \cup \{ct\}$;
5. $E_H := E_H \cup \{(work(t-1), work(t))\}$;

$Back\_A()$:

1. *delete* $work(t)$;
2. $t := t - 1$;

$Stop\_A()$:

1. $STOP := 1$;

$Forward\_BE\_A()$:

1. $E_H := E_H \cup \{(work(t), work(t-i))\}$;

$Back\_BE\_A()$:

1. $i := i + 1$;

$Recognized\_BE\_A()$:

1. $i := 0$;

4. Properties of the graph exploration algorithm

Theorem 1. *By performing an exploration algorithm on the graph G, the agent explores the graph with accuracy to isomorphism.*

Proof. By executing the specified algorithm, provided that $n \geqslant 3$ the agent-researcher will execute the $FORWARD(v)$ procedure at least twice, during which new white nodes of the explored graph G will be visited. After one execution of this procedure, a new node of graph H is added to the agent-experimenter's memory. Thus, while executing the algorithm, an implicit numbering $\varphi\colon V_G \to V_H$ is created, where the equality $\varphi(v) = ct$ is established when node v is coloring black. This numbering is a bijection because, in a connected graph G, all nodes are reachable from the starting node and, therefore, will be visited by the agent-researcher.

Since the algorithm is based on a depth-first traversal method, all edges of the explored graph are divided into tree edges and back edges. During the execution of the $FORWARD(v)$ procedure, the agent-researcher also identifies a tree edge (v, u) and numbers node u, such that the edge (v, u) of graph G uniquely corresponds to the edge $(\varphi(v), \varphi(u))$ in graph H. During the execution of the $EXPL\_IE(v)$ procedure, the agent identifies the back edges (v, u) of graph G and matches them uniquely with edges $(\varphi(v), \varphi(u))$ of graph H. Thus, the mapping φ is an isomorphism from graph G to graph H. ■

Let us calculate the time and space complexities of the proposed algorithm. We will also determine an upper bound on the number of edge transitions the agent-researcher needs to make in order to fully explore the studied graph G.

During the description of the algorithm, we mentioned that the agent-researcher has a working path, which is the path from the agent's starting node to its current working node. In other words, at each step of the algorithm, the working path is a simple path from the starting node v, numbered $\varphi(v) = 1$, to the node u, numbered $\varphi(u) = ct$. Thus, it can be concluded that the length of the working path does not exceed n. Executing the $FORWARD(v)$ and $BACK(v)$ procedures, the agent-researcher takes one step while traversing one edge. When executing the $EXPL\_IE(v)$ procedure to explore back edges from a node, the agent-researcher traverses no more than $n - 1$ edges of the working path backward and explores no more than $n - 2$ back edges.

When calculating the time complexity of the constructed exploration algorithm, we will assume that the agent-researcher's transition from one node to another takes time equal to some constant. It is obvious that the total time for analyzing the vicinity of the working node $Q(v)$ and selecting the necessary edges is bounded above as $O(n^2)$. We will also assume that processing a single message by the agent-experimenter takes no longer than the time it takes for the agent-researcher to move from one node to another. Considering this, the relationships that define the time complexity of the algorithm will be as follows:

1. The $FORWARD(v)$ procedure is executed no more than $n - 1$ times, and the total time for its execution is evaluated as $O(n)$.
2. The $BACK(v)$ procedure is executed no more than $n-1$ times, and the total time for its execution is evaluated as $O(n)$.
3. The $EXPL\_IE(v)$ procedure is executed no more than $n - 2$ times, each taking no more than $2 \times n + 2 \times (n - 2)$ time units, so the total execution time for the procedure is evaluated as $(2 \times n + 2 \times (n - 2)) \times (n - 2)$, which is $O(n^2)$, considering the steps spent on placing and picking up the stone.
4. The $STOP(v)$ procedure is executed once, and its asymptotic complexity is $O(1)$.

Thus, the total number of edge transitions made by the agent-researcher does not exceed $(n-1)+(n-1)+(2\times(n-1)+2\times(n-2))\times(n-2)$, meaning the upper bound for the number of transitions satisfies the relation: $M(n) = O(n^2)$.

The overall time complexity of the proposed algorithm satisfies the relation: $T(n) = O\left(n^2\right)$.

The space complexity $S(n)$ of the proposed algorithm is determined by the complexities of the lists V_H, E_H, $work(1), \ldots, work(t)$, whose complexities are defined by the magnitudes $O(n)$, $O(n^2)$, $O(n)$ respectively, and therefore $S(n) = O(n^2)$.

At each step of the algorithm, the agent-researcher can send at most one message to the agent-experimenter. Since the message contains no information about the graph elements, its size can be considered constant. Thus, the volume of transmitted information is estimated as $O(n^2)$. Therefore, $K(n) = O(n^2)$.

Considering the above, the following theorem holds.

Theorem 2. *The time, space, and communication complexities of the exploration algorithm are $O(n^2)$, and the upper bound on the number of edge transitions made by the agent-researcher is $O(n^2)$. At the same time, the algorithm uses one color and one stone.*

5. Conclusions

The study addresses the problem of graph exploration by a collective of agents. A new algorithm for exploring simple, undirected, finite graphs is proposed, with quadratic time, space, and communication complexities, and an upper bound on the number of edge transitions made by the agent-researcher, which equals $O(n^2)$. The agent-researcher has finite memory that does not depend on the dimensionality of the explored graph. The agent-experimenter, at the same time, has finite memory at each step and unbounded growing internal memory, the size of which depends on the dimensionality of the explored graph. To execute the graph exploration algorithm, the collective of agents requires only one color and one stone.

[1] Glushkov V. M., Letichevsky A. A. Theory of Discrete Converters (Selected Topics of Algebra and Logic). Nauka. 5–20 (1973).

[2] Stopkin A. V. Finite graph exploration by a mobile agent. Mathematical Modeling and Computing. **12** (1), 75–82 (2025).

[3] Kuipers B. The spatial semantic hierarchy. Artificial Intelligence. **119** (1–2), 191–233 (2000).

[4] Fraigniaud P., Jecincas D., Peer G., Pelc A., Peleg D. Graph Exploration by a Finite Automaton. Mathematical Foundations of Computer Science 2004 (MFCS 2004). 451–462 (2004).

[5] Das S., Flocchini P., Kutten S., Nayak A., Santoro N. Map construction of unknown graphs by multiple agents. Theoretical Computer Science. **385** (1–3), 34–48 (2007).

[6] Wang H., Jenkin M., Dymond P. It can be beneficial to be 'lazy' when exploring graph-like worlds with multiple robots. Proceedings of the IASTED International Conference on Advances in Computer Science and Engineering (ACSE). 55–60 (2009).

[7] Nagavarapu S. C., Vachhani L., Sinha A., Buriuly S. Generalizing Multi-agent Graph Exploration Techniques. International Journal of Control, Automation and Systems. **19**, 491–504 (2020).

[8] Stopkin A. V. Algorithm for exploration of a simple undirected graph by a collective of agents. Scientific notes of Taurida National V. I. Vernadsky University. Series: Technical Sciences. **35** (74 (5)), 303–309 (2024).

[9] Stopkin A. V. Multi-agent system for non-oriented graphs exploration. Information Technology and Society. **3** (14), 38–43 (2024).

[10] Stepkin A. Using a Collective of Agents for Exploration of Undirected Graphs. Cybernetics and Systems Analysis. **51** (2), 223–233 (2015).

[11] Selden M., Zhou J., Campos F., Lambert N., Drew D., Pister K. S. J. BotNet: A Simulator for Studying the Effects of Accurate Communication Models on Multi-Agent and Swarm Control. 2021 International Symposium on Multi-Robot and Multi-Agent Systems (MRS). 101–109 (2021).

[12] Li D., Ge S. S., He W., Ma G., Xie L. Multilayer formation control of multi-agent systems. Automatica. **109**, 108558 (2019).

[13] Zhang T., Ma X., Li H., Wang Z., Xie S., Luo J. Ordered-Bipartite Consensus of Multi-Agent Systems under Finite Time Control. Applied Sciences. **12** (23), 12337 (2022).

Розпізнавання простих графів колективом агентів

Стьопкін А. В.

ДВНЗ "Донбаський державний педагогічний університет",
вул. Наукова, 13, 49020, Дніпро, Україна

В статті розглядається проблема розпізнавання простих неорієнтованих графів мультиагентною системою, що складається з двох агентів: агента-дослідника, який може рухатись графом, зчитувати та змінювати помітки елементів графа та обмінюватись інформацією з другим агентом — агентом-експериментатором, який і будує мапу досліджуваного графу в своїй пам'яті у вигляді списків ребер та вершин. Запропоновано алгоритм розпізнавання квадратичної (від числа вершин графа) часової, ємнісної та комунікаційної складностей. Число переходів по ребрах, які треба здійснити агентам-дослідникам оцінюється як $O(n^2)$. Для роботи алгоритму використовується фарба одного кольору та один камінь. Метод базується на методі обходу графа в глибину.

Ключові слова: *розпізнавання графа; мультиагентна система; обхід графа; складність алгоритму; обхід в глибину.*